



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №2
Технології розробки програмного забезпечення
«ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ
ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ.
КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»
Варіант 23

Виконав:
студент групи ІА-24
Красношапка Р. О.
Перевір:
Мягкий М. Ю.

Київ 2024

Тема: ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ. КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ

Мета: Розробити діаграми варіантів використання та класів.

Хід роботи:

Project Management software (proxy, chain of responsibility, abstract factory, bridge, flyweight, client-server) Програмне забезпечення для управління проектами повинно мати наступні функції: супровід завдань/вимог/проектів, списків команд, поточних завдань, планування за методологіями agile/kanban/rup (включаючи дошку завдань, ітерації тощо), мати можливість прикріплювати вкладені файли до завдань та посилатися на конкретні версії програми, зберігати виконувані файли для кожної версії.

Діаграми варіантів використання (Use-case diagram):

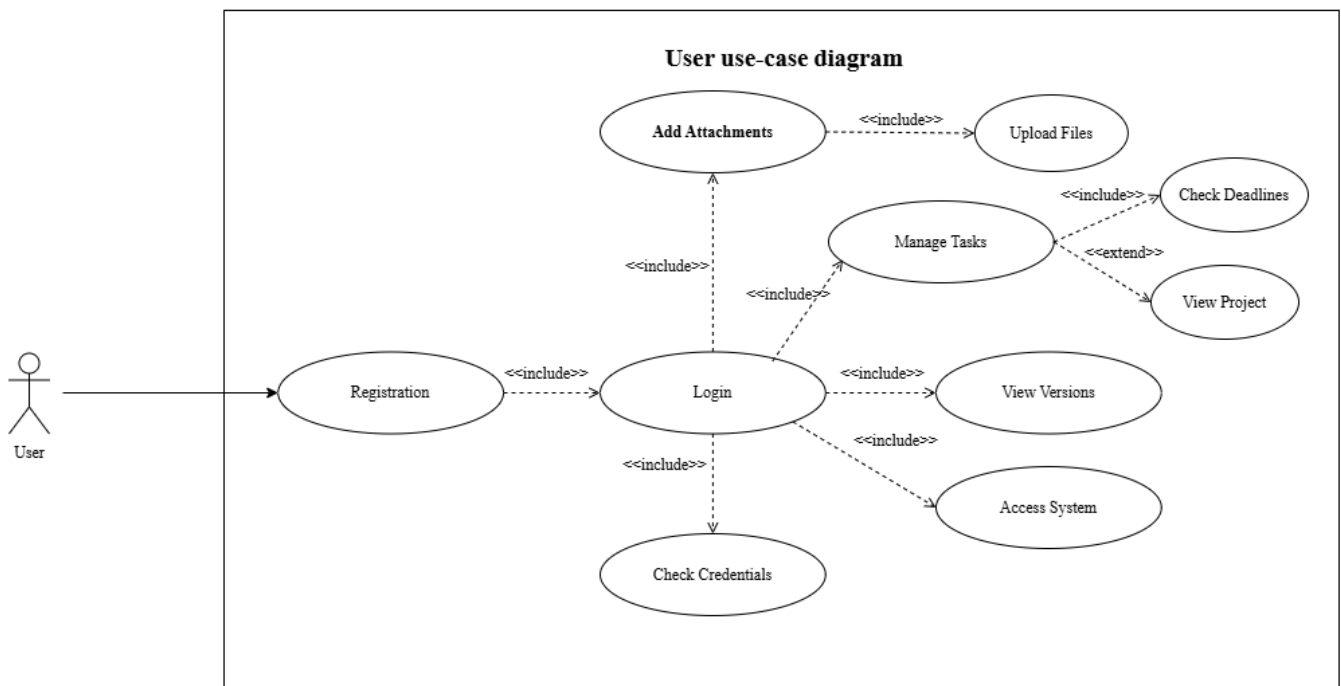


Рис. 1.1 - Use-case diagram для User

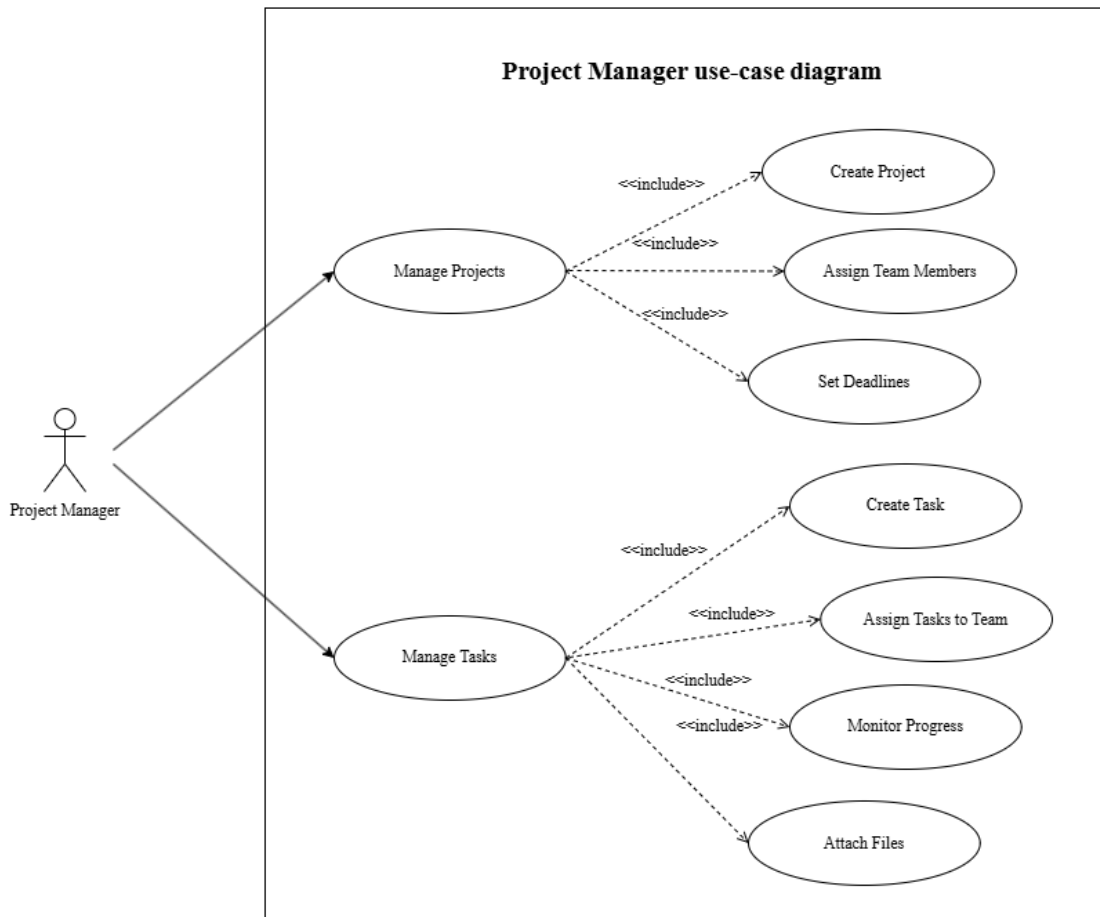


Рис. 1.2 - Use-case diagram для РМ

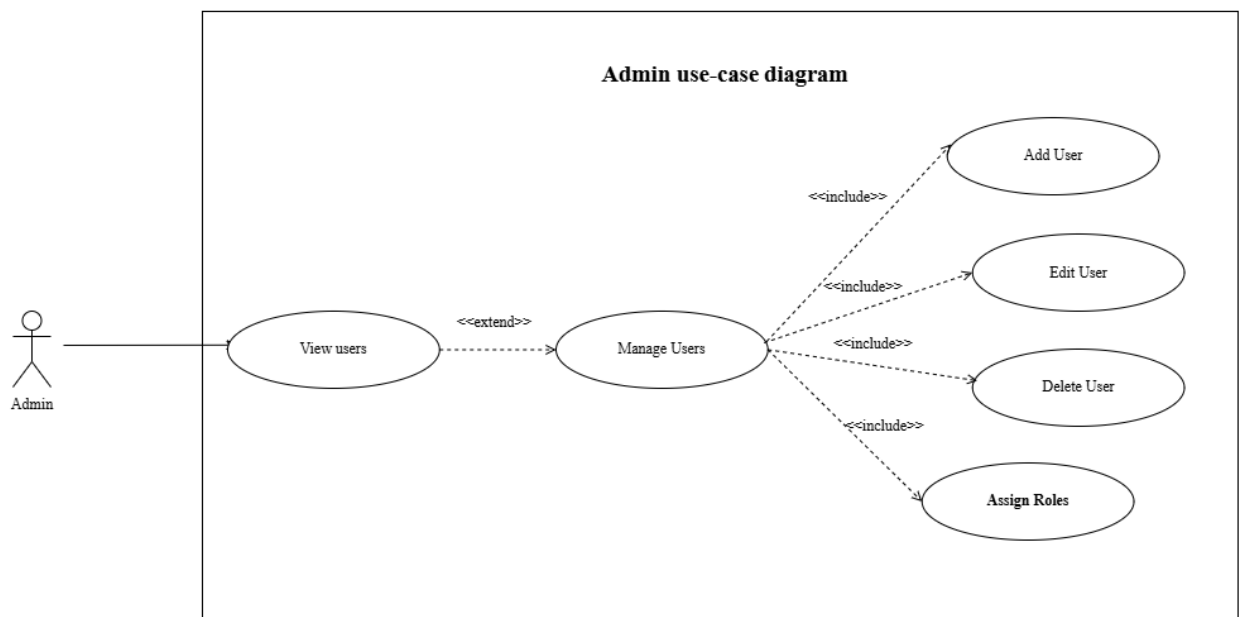


Рис. 1.3 - Use-case diagram для Admin

Варіанти використання (прецеденти)

1) Додавання та редагування завдань у проекті

Передумови: Користувач авторизований у системі

Постумови: Завдання успішно додано або відредаговано у проекті

Взаємодіючі сторони: Користувач, система управління проектами

Короткий опис: Цей варіант описує процес додавання нового завдання або редагування існуючого завдання в проекті.

Основний потік подій:

1. Користувач відкриває проект у системі управління проектами.
2. Користувач натискає кнопку "Додати завдання" або обирає існуюче завдання для редагування.
3. Користувач заповнює необхідні поля, такі як опис, пріоритет, термін виконання та відповідальні особи.
4. Користувач натискає кнопку "Зберегти". Система зберігає завдання і підтверджує успішне додавання/редагування.
5. Користувач отримує сповіщення про успішне виконання дій.

• **Винятки:**

Виняток №1: Пропущені обов'язкові поля. Якщо користувач не заповнив усі обов'язкові поля, система відображає відповідне повідомлення з деталями про помилку.

2) Прикріплення файлів до завдань

Передумови: Завдання вже існує у проекті

Постумови: Файл успішно прикріплений до завдання

Взаємодіючі сторони: Користувач, система управління проектами

Короткий опис: Даний варіант описує процес прикріплення файлів до завдання для зручності доступу та обміну інформацією.

Основний потік подій:

1. Користувач відкриває завдання у проекті.
2. Користувач натискає кнопку "Прикріпити файл".
3. Відкривається діалогове вікно для вибору файлу з комп'ютера.
4. Користувач вибирає файл і підтверджує вибір.
5. Система прикріплює файл до завдання та підтверджує успішне прикріплення.

Винятки:

- **Виняток №1:** Файл занадто великого розміру. Якщо розмір файлу перевищує максимальний допустимий, система виводить відповідне повідомлення про помилку.

3) Перегляд версій проекту

Передумови: Проект має кілька версій

Постумови: Користувач отримує інформацію про доступні версії проекту

Взаємодіючі сторони: Користувач, система управління проектами

Короткий опис: Цей варіант описує процес перегляду доступних версій проекту для можливості вибору та роботи з конкретною версією.

Основний потік подій:

1. Користувач відкриває проект у системі управління проектами.
2. Користувач обирає вкладку "Версії".
3. Система відображає список доступних версій проекту з інформацією про номер версії, дату випуску та примітки.
4. Користувач може вибрати конкретну версію для перегляду або завантаження.

Винятки:

- **Виняток №1:** Відсутні версії проекту. Якщо версії не знайдено, система відображає повідомлення про відсутність версій.

UML діаграма класів

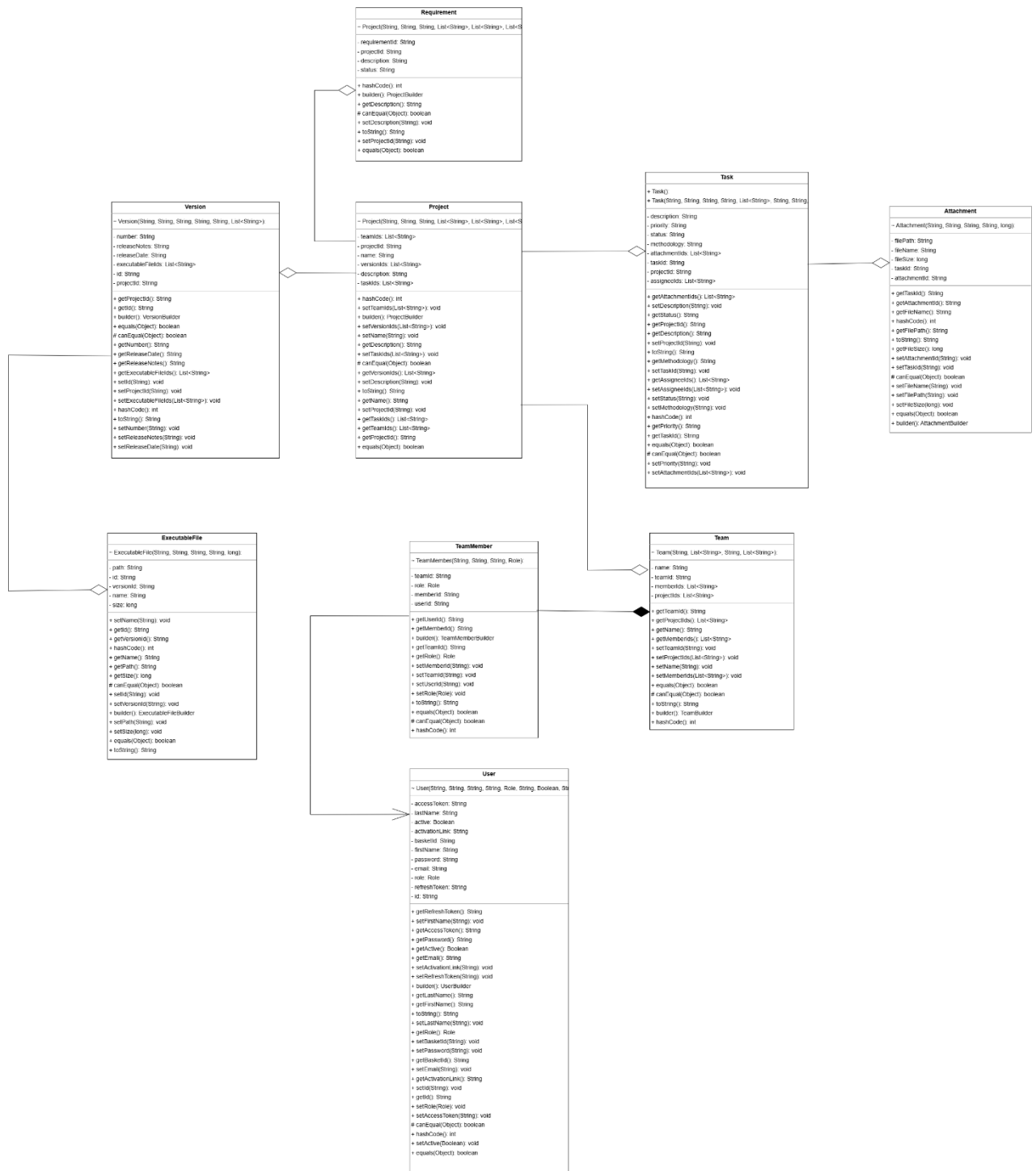


Рис. 2.1 – UML діаграма класів

Зображення структури бази даних

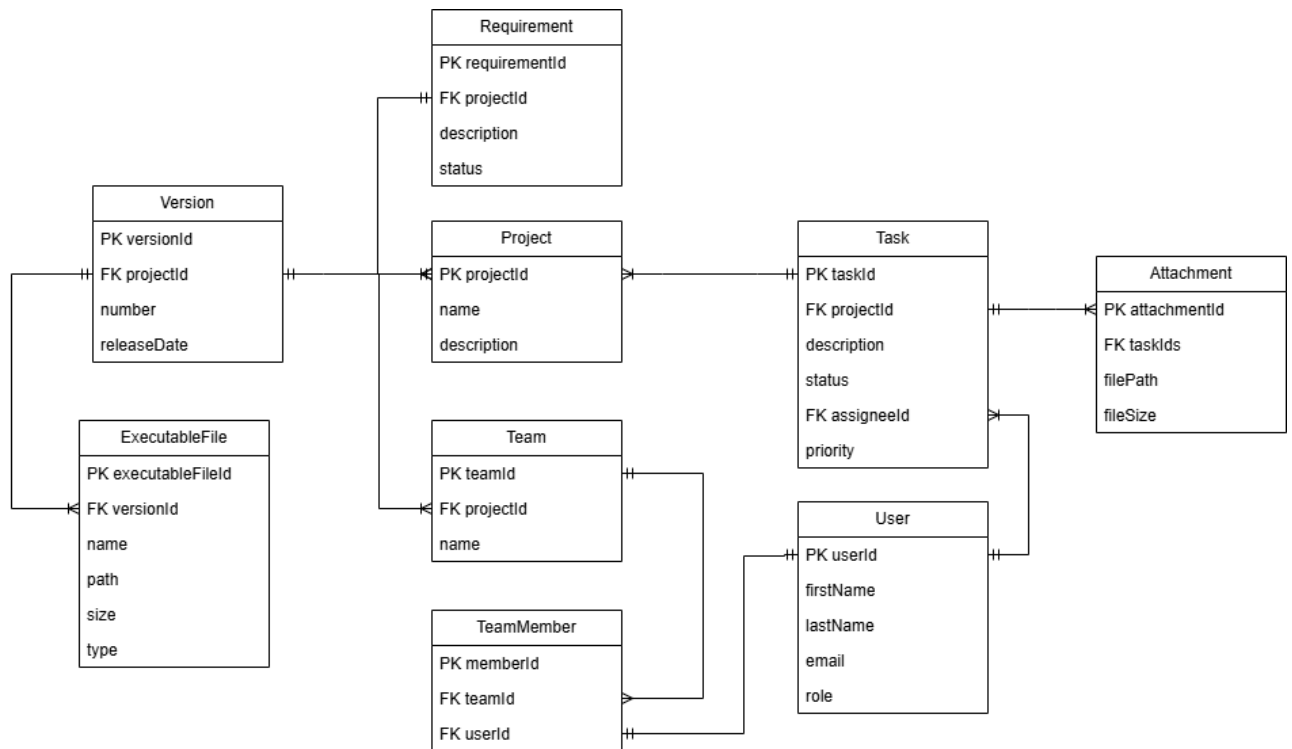


Рис. 3.1 – зображення структури бази даних

Реалізація

<https://github.com/QUIRINO228/projectManagement>

Висновок: при виконанні цієї лабораторної роботи я розробив діаграми використання, класів, а також схему бази даних.